


# Segmentation in 2D and 3D Image Using Tissue-Like P System

View metadata, citation and similar papers at [core.ac.uk](http://core.ac.uk)

brought to you by  CORE

provided by idUS. Depósito de Investigación Universidad de Sevilla

and Pedro Real Jurado<sup>1</sup>

<sup>1</sup> Research Group on Computational Topology and Applied Mathematics

University of Sevilla

Avda. Reina Mercedes s/n, 41012, Sevilla, Spain

<sup>2</sup> Karunya University

Coimbatore, Tamilnadu, India

{hepzi,sbdani,real}@us.es

**Abstract.** Membrane Computing is a biologically inspired computational model. Its devices are called P systems and they perform computations by applying a finite set of rules in a synchronous, maximally parallel way. In this paper, we open a new research line: P systems are used in Computational Topology within the context of the Digital Image. We choose for this a variant of P systems, called *tissue-like P systems*, to obtain in a general maximally parallel manner the segmentation of 2D and 3D images in a constant number of steps. Finally, we use a software called *Tissue Simulator* to check these systems with some examples.

## 1 Introduction

Natural Computing studies new computational paradigms inspired from Nature. It abstracts the way in which Nature “computes”, conceiving new computing models. There are several fields in Natural Computing that are now well established as are Genetic Algorithms ([8]), Neural Networks ([10]), DNA-based molecular computing ([1]).

Membrane Computing is a theoretical model of computation inspired by the structure and functioning of cells as living organisms able to process and generate information. The computational devices in Membrane Computing are called *P systems* [15]. Roughly speaking, a P system consists of a membrane structure, in the compartments of which one places multisets of objects which evolve according to given rules. In the most extended model, the rules are applied in a synchronous non-deterministic maximally parallel manner, but some other semantics are being explored.

According to their architecture, these models can be split into two sets: cell-like P systems and tissue-like P systems [19]. In the first systems, membranes are hierarchically arranged in a tree-like structure. The inspiration for such architecture is the set of vesicles inside the cell. All of them perform their biological processes in parallel and life is the consequence of the harmonious conjunction

of such processes. This paper is devoted to the second approach: tissue-like P systems.

Segmentation in computer vision (see [9]), refers to the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

There exists different techniques to segment an image. Few techniques are Clustering methods [7], Histogram-based methods [12], Watershed transformation methods [23], Graph partitioning methods [22].

Some of the practical applications of image segmentation are like Medical Imaging [7], Study of Anatomical Structure, Locate objects in Satellite Images (roads, forests, etc.) [11], Face Recognition [21].

J. Chao and J. Nakayama connected in [4] Natural Computing and Algebraic Topology using Neural Networks (extended Kohonen mapping). We use for the first time, the power and efficiency of a variant of P systems called tissue-like P systems (see [5]) to segment the image in 2D.

The paper is structured as follows: in the next section we present the definition of basic tissue like P systems with input and show an example to understand how these systems work. In section 3, we design a family of systems for edge-based segmentation in 2D image. After, we check our model using a software called tissue simulator with two images very easy. At the end of this section, we introduce a family of tissue-like P systems to obtain an edge-based segmentation of 3D images. Finally, some conclusions and future work are given in the last section.

## 2 Description of a Model of Membranes

Membrane computing models was first presented by Martín-Vide *et al.* in [13] and it has two biological inspirations (see [14]): intercellular communication and cooperation between neurons, but in this paper we work with a variant presented in [19] *with cell division* and the system which is presented by Díaz-Pernil presented in [6] a formalization of *Tissue-like P systems* (without cellular division). The common mathematical model of these two mechanisms is a network of processors dealing with symbols and communicating these symbols along channels specified in advance.

The main features of this model, from the computational point of view, are that cells have not polarizations (the contrary holds in the cell-like model of P systems, see [16]) and the membrane structure is a general graph.

Formally, a *tissue-like P system* with input of degree  $q \geq 1$  is a tuple

$$\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, \dots, w_q, \mathcal{R}, i_\Pi, o_\Pi),$$

where

1.  $\Gamma$  is a finite *alphabet*, whose symbols will be called *objects*,  $\Sigma(\subset \Gamma)$  is the input alphabet,  $\mathcal{E} \subseteq \Gamma$  (the objects in the environment),
2.  $w_1, \dots, w_q$  are strings over  $\Gamma$  representing the multisets of objects associated with the cells at the initial configuration,
3.  $\mathcal{R}$  is a finite set of communication rules of the following form:  $(i, u/v, j)$ , for  $i, j \in \{0, 1, 2, \dots, q\}, i \neq j, u, v \in \Gamma^*$ ,
4.  $i_\Pi, o_\Pi \in \{0, 1, 2, \dots, q\}$ .

A tissue-like P system of degree  $q \geq 1$  can be seen as a set of  $q$  cells (each one consisting of an elementary membrane) labelled by  $1, 2, \dots, q$ . We will use 0 to refer to the label of the environment,  $i_\Pi$  denotes the input region and  $o_\Pi$  denotes the output region (which can be the region inside a cell or the environment).

The strings  $w_1, \dots, w_q$  describe the multisets of objects placed in the  $q$  cells of the system. We interpret that  $\mathcal{E} \subseteq \Gamma$  is the set of objects placed in the environment, each one of them available in an arbitrary large amount of copies.

The communication rule  $(i, u/v, j)$  can be applied over two cells labelled by  $i$  and  $j$  such that  $u$  is contained in cell  $i$  and  $v$  is contained in cell  $j$ . The application of this rule means that the objects of the multisets represented by  $u$  and  $v$  are interchanged between the two cells. Note that if either  $i = 0$  or  $j = 0$  then the objects are interchanged between a cell and the environment.

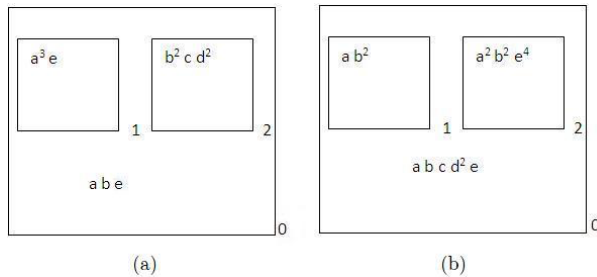
Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered). In one step, each object in a membrane can only be used for one rule (non-deterministically chosen when there are several possibilities), but any object which can participate in a rule of any form must do it, i.e, in each step we apply a maximal set of rules.

Now, to understand how we can obtained a computation of one of these P systems we present an example of them:

Consider us the tissue-like P system  $\Pi' = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$  where

1.  $\Gamma = a, b, c, d, e, \Sigma = \emptyset, \mathcal{E} = a, b, e,$
2.  $w_1 = a^3 e, w_2 = b^2 c d,$
3.  $\mathcal{R} = \{(1, a/b, 2), (2, c/b^2, 0), (2, d/e^2, 0), (1, e/\lambda, 0)\},$
4.  $i_\Pi = 1$  and  $o_\Pi = 0$ .

We can observe the initial configuration of this system in the figure 1 (a). We have four rules to apply, and after applying the rules the next configuration is



**Fig. 1.** (a) Initial Configuration of system  $\Pi'$  (b) Following Configuration of  $\Pi'$

shown in (b), the system apply one time each one of them. If reader looks at the elements in the environment one can observe the number of the copies of the elements  $a, b, e$  always are one, because they are the objects that appear in the environment initially (we have an arbitrary large amount of copies of them), but  $d$  has two copies because it is not an initial element of the environment. Usually, the elements of the environment are not described in the system to better understanding of the configurations of this.

### 3 Segmenting Digital Images in Constant Time

In this section, we segment images based on edge-based segmentation. Edge-based segmentation finds boundaries of regions which are sufficiently different from each other. We define two family of tissue-like P systems for edge-based segmentation, one of them to segment 2D images and after, we adapt these systems to segment 3D images.

#### 3.1 A Family of Tissue-Like P Systems for a 2D Segmentation

We can divide the image in multiple pixels forming a network of points of  $\mathbb{N}^2$ . Let  $\mathcal{C} \subseteq \mathbb{N}$  be the set of all colors in the given 2D image and they are in a certain order. Moreover, we will suppose each pixel is associated with a color of the image. Then we can codify the pixel  $(i,j)$  with associated color  $a \in \mathcal{C}$  by the object  $a_{ij}$ .

The following question is the adjacency problem. We have decided to use in this paper the 4-adjacency [2,3].

In this point, we want to find the border cells of the different color regions present in the image. Then, for each image with  $n \times m$  pixels ( $n, m \in \mathbb{N}$ ) we will construct a tissue-like P system whose input is given by the objects  $a_{ij}$  codifying a pixel, with  $a \in \mathcal{C}$ . The output of the system is given by the objects that appear in the output cell when it stops.

So, we can define a family of tissue-like P systems to do the edge-based segmentation to 2D images. For each  $n, m \in \mathbb{N}$  we consider the tissue-like P system  $\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$ , defined as follows:

- (a)  $\Gamma = \Sigma \cup \{\bar{a}_{ij} : 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{A_{ij} : 1 \leq i \leq n, 1 \leq j \leq m, A \in \mathcal{C}\}$ ,  $\Sigma = \{a_{ij} : a \in \mathcal{C}, 1 \leq i \leq n, 1 \leq j \leq m\}$ ,  $\mathcal{E} = \Gamma - \Sigma$ ,
- (b)  $w_1 = w_2 = \emptyset$ ,
- (c)  $R$  is the following set of communication rules:

1.  $(1, a_{ij}b_{kl}/\bar{a}_{ij}A_{ij}b_{kl}, 0)$ , for  $a, b \in \mathcal{C}$ ,  $a < b$ ,  $1 \leq i, k \leq n$ ,  $1 \leq j, l \leq m$  and  $(i, j), (k, l)$  adjacents.

These rules are used when image has two adjacent pixels with different associated colors(border pixels), and the pixel with less associated color is marked and system brings from the environment an object representing this marked pixel(edge pixel).

2.  $(1, \bar{a}_{ij}a_{ij+1}\bar{a}_{i+1j+1}b_{i+1j} / \bar{a}_{ij}\bar{a}_{ij+1}A_{ij+1}\bar{a}_{i+1j+1}b_{i+1j}, 0)$  for  $a, b \in \mathcal{C}, a < b, 1 \leq i \leq n-1, 1 \leq j \leq m-1$ .  
 $(1, \bar{a}_{ij}a_{i-1j}\bar{a}_{i-1j+1}b_{ij+1} / \bar{a}_{ij}\bar{a}_{i-1j}A_{i-1j}\bar{a}_{i-1j+1}b_{ij+1}, 0)$  for  $a, b \in \mathcal{C}, a < b, 2 \leq i \leq n, 1 \leq j \leq m-1$ .  
 $(1, \bar{a}_{ij}a_{ij+1}\bar{a}_{i-1j+1}b_{i-1j} / \bar{a}_{ij}\bar{a}_{ij+1}A_{ij+1}\bar{a}_{i-1j+1}b_{i-1j}, 0)$  for  $a, b \in \mathcal{C}, a < b, 2 \leq i \leq n, 1 \leq j \leq m-1$ .  
 $(1, \bar{a}_{ij}a_{i+1j}\bar{a}_{i+1j+1}b_{ij+1} / \bar{a}_{ij}\bar{a}_{i+1j}A_{i+1j}\bar{a}_{i+1j+1}b_{ij+1}, 0)$  for  $a, b \in \mathcal{C}, a < b, 1 \leq i \leq n-1, 1 \leq j \leq m-1$ .

The rules mark with a bar the pixels which are adjacent to two same color pixels and which were marked before, but with the condition that the marked objects are adjacent to an other pixel with a different color. Moreover, an edge object representing the last marked pixel is brought from the environment.

3.  $(1, A_{ij}/\lambda, 2)$ , for  $1 \leq i \leq n, 1 \leq j \leq m$ .

This rule is used to send the edge pixels to the output cell.

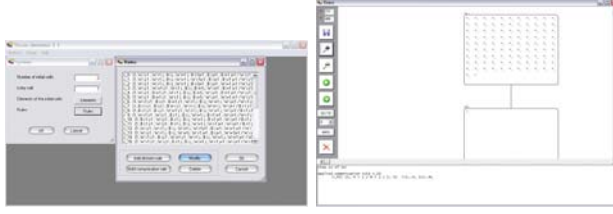
d)  $i_{\Pi} = 1, o_{\Pi} = 2$ .

**An overview of the Computation:** Input objects  $a_{ij}$  codifying the colored pixels from an 2D image appear in the input cell and with them the system begins to work. Rules of type 1, in a parallel manner, identify the border pixels and bring the edge pixels from the environment. These rules need 4 steps to mark all the border pixels. From the second step, the rules of type 2 can be used with the first rules at the same time. So, in other 4 steps we can bring from the environment the edge pixels adjacent to two border pixels as we explain above. System can apply the first two types of rules simultaneously in some configurations, but it always applies the same number of these two types of rules because this number is given by edge pixels (we consider 4-adjacency). Finally, the third type of rules are applied in the following step on edge pixels appear in the cell. So, with one step more we will have all the edge pixels in the output cells. Thus, we need only 9 steps to obtain an edge-based segmentation for an  $n \times m$  image. Then, we can conclude the problem of edge-segmentation in 2D images is resolved in this paper in a constant time respect to the number of steps of any computation.

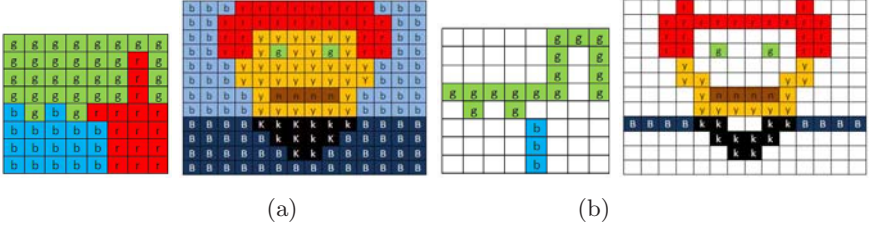
### 3.2 Checking This Family of Systems with Tissue Simulator

We have used a software called *tissue-simulator* (See section 2) introduced by Borrego-Ropero et al. in [20]. We have simulated our family of systems to segment 2D images with this software. Finally, we have introduced as instances of our system the examples that appear in 3 (a) and, in a constant number of steps we have obtained a codifying of the edge-segmentation (that appear in 3 (b)) of the examples introduced before.

We consider, in a first case an  $8 \times 8$  image, and the order of the colors used in this image is the following: green, blue and red. In a second case we work with an image of size  $12 \times 14$ . In this example, we take the colors in the following order: Red, green, brown, orange, black, blue and light blue.



**Fig. 2.** Two images about Tissue Simulator



**Fig. 3.** (a) Input images (b) Edge-segmentation of images

r-red; b-blue; B-dark blue; g-green; y-yellow; n-brown; k-black; blank-white.

### 3.3 Segmenting 3D Images

The following step to consists to extend our models to work with 3D images. Now, the input dates are voxels  $((i, j, k) \in \mathbb{N}^3)$  that the are codifying by the elements  $a_{ijk}$ , with  $a \in \mathcal{C}$ . We use here 26-adjacency relationship between voxels. Then, we can define a family of tissue-like P systems. For each  $n, m \in \mathbb{N}$  we consider the  $\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$  to do an edge-based segmentation to a 3D image as follows

- (a)  $\Gamma = \Sigma \cup \{A_{ijk} : 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq l, a \in \mathcal{C}\}, \Sigma = \{a_{ijk} : a \in \mathcal{C}, 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq l\}, \mathcal{E} = \Gamma - \Sigma,$
- (b)  $w_1 = w_2 = \emptyset,$
- (c)  $R$  is the following set of rules:
  1.  $(1, a_{i_1 j_1 k_1} b_{i_2 j_2 k_2} / A_{i_1 j_1 k_1} b_{i_2 j_2 k_2}, 0),$  for  $1 \leq i_1, i_2 \leq n, 1 \leq j_1, j_2 \leq m, 1 \leq k_1, k_2 \leq l, (i_1, j_1, k_1)$  and  $(i_2, j_2, k_2)$  adjacents voxels and finally,  $a, b \in \mathcal{C}$  with  $a < b$ .  
These rules are used when image has two adjacent border voxels. Then, system brings from the environment an object representing the voxel with less associated color (edge voxel).
  2.  $(1, A_{ijk} / \lambda, 2),$  for  $1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq l.$   
These rules are used to send the edge voxels to the output cell.
- d)  $i_\Pi = 1, o_\Pi = 2.$

**An overview of the Computation:** This computation would be very similar if we consider an 26-adjacency in 3D. Rules of type 1 identify the border pixels and bring the edge pixels from the environment. These rules need as much 26

steps for this. Finally, the second type of rules are applied in the following step and send the edge pixels to the output cell. So, we need again a constant amount of steps to resolve the edge-segmentation problem in 3D.

## 4 Conclusions and Future Work

It is shown in this paper, if we consider a 4-adjacency, a segmentation of a 2D image can be given in a constant number of steps using tissue-like P systems.

With this paper new research lines have been opened. We can work in some of them directly, define new systems to obtain other homological informations (spanning trees, homology gradient vector field, representative cocycles of cohomology generators, etc) for 2D or 3D images. But, other lines need more time and deep research work, as are: to develop an efficient sequential software using these techniques, to develop an efficient software working with a cluster. Moreover, both of them could be applied in different areas as are: medical imaging, locate objects in satellite in satellite images, etc.

## Acknowledgement

The first author acknowledge the support of the project "Computational Topology and Applied Mathematics" PAICYT research project FQM-296. The second author acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, and the support of the Project of Excellence of the Junta de Andalucía, grant P08-TIC-04200. The third author acknowledge the support of the project MTM2006-03722 of the Ministerio español de Educación y Ciencia and the project PO6-TIC-02268 of Excellence of Junta de Andalucía.

## References

1. Adleman, L.M.: Molecular computations of solutions to combinatorial problems. *Science* 226, 1021–1024 (1994)
2. Rosenfeld, A.: Digital topology. *American Mathematical Monthly* 86, 621–630 (1979)
3. Rosenfeld, A.: Connectivity in Digital pictures. *Journal for Association of Computing Machinery* 17(1), 146–160 (1970)
4. Chao, J., Nakayama, J.: Cubical Singular Simplex Model for 3D Objects and Fast Computation of Homology Groups. In: *Proceedings of ICPR 1996 IEEE*, pp. 190–194 (1996)
5. Díaz-Pernil, D., Gutiérrez, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: A uniform family of tissue P systems with cell division solving 3-COL in a linear time. *Theoretical Computer Science* 404, 76–87 (2008)
6. Díaz-Pernil, D., Pérez-Jiménez, M.J., Romero, A.: Efficient simulation of tissue-like P systems by transition cell-like P systems. *Natural Computing*, <http://dx.doi.org/10.1007/s11047-008-9102-z>

7. Wang, D., Lu\*, H., Zhang, J., Liang, J.Z.: A Knowledge-Based Fuzzy Clustering Method with Adaptation Penalty for Bone Segmentation of CT images. In: Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pp. 6488–6491 (2005)
8. Holland, J.H.: *Adaptation in Natural and Artificial Systems* (1975)
9. Shapiro, L.G., Stockman, G.C.: *Computer Vision* (2001)
10. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133 (1943)
11. Sharmay, O., Miocz, D., Antony, F.: Polygon Feature Extraction from Satellite imagery based on color image segmentation and medial axis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVII*, 235–240 (2008)
12. Tobias, O.J., Seara, R.: Image Segmentation by Histogram Thresholding Using Fuzzy Sets. *IEEE Transactions on Image Processing* 11(12), 1457–1465 (2002)
13. Martín-Vide, C., Pazos, J., Păun, G., Rodríguez-Patón, A.: A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. In: Ibarra, O.H., Zhang, L. (eds.) *COCOON 2002. LNCS*, vol. 2387, pp. 290–299. Springer, Heidelberg (2002)
14. Martín-Vide, C., Pazos, J., Păun, G., Rodríguez Patón, A.: Tissue P systems. *Theoretical Computer Science* 296, 295–326 (2003)
15. Păun, G.: Computing with membranes. *Journal of Computer and System Sciences* 61, 108–143 (2000)
16. Păun, G.: *Membrane Computing: An Introduction*. Springer, Heidelberg (2002)
17. Păun, A., Păun, G.: The power of communication: P systems with symport/antiport. *New Generation Computing* 20(3), 295–305 (2002)
18. Păun, G.: Computing with Membranes: Attacking **NP**-complete Problems. In: *Unconventional Models of Computation, UMC'2K*, pp. 94–115. Springer, Heidelberg (2000)
19. Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Tissue P System with cell division. In *Second Brainstorming Week on Membrane Computing, Sevilla, Report RGNC 01/2004*, pp. 380–386 (2004)
20. <http://www.tissuesimulator.es.kz>
21. Kim, S.-H., Kim, H.-G., Tchah, K.-H.: Object oriented face detection using colour transformation and range segmentation. *IEEE Electronics Letters* 34, 979–980 (1998)
22. Xiaojing, Y., Ning, S., George, Z.: A narrow band graph partitioning method for skin lesion segmentation. *Elsevier Science Pattern Recognition* 42(6), 1017–1028 (2009)
23. Yazid, H., Arof, H.: Image Segmentation using Watershed Transformation for Facial Expression Recognition. *IFMBE Proceedings, 4th Kuala Lumpur International Conference on Biomedical Engineering* 21, 575–578 (2008)